ALGORITHM I

RDPSO ALGORITHM

	Main Program Loop		***Evolve Swarm Algorithm***
1	For each swarm in the collection	1	For each particle $\boldsymbol{n}$ in the swarm $\boldsymbol{s}$
		2	Update particles' objective function $\boldsymbol{f}(\boldsymbol{x_n}[t])$
2	Evolve the swarm ***Evolve Swarm Algorithm*** (right →)	3	Update particles' best
		4	Move particle
3	Allow the swarm to spawn a new swarm from the $\boldsymbol{n_I}$ best performing robots in the socially excluded group	5	If swarm $\boldsymbol{s}$ gets better
		6	Reward swarm $\boldsymbol{s}$ with the best performing robot in the socially excluded group
4	Move "failed" swarms to the socially excluded group	7	If swarm $\boldsymbol{s}$ has not improved
		8	Punish swarm $\boldsymbol{s}$ by excluding the worst performing robot adding it to a socially excluded group

The key issue in this novel approach is the answer to the question: What robots belonging to the socially excluded group do? In fact the answer is the same that we would give if asking about a group excluded from our society: they do not do "anything". Instead of searching for the objective function's global optimum (*i.e.*, the main activity of the society) like the other robots in the active swarms do, they basically randomly wander in the scenario. Note, however, that they are always aware of their individual solution and the global solution of the socially excluded group.

Consider the example in Fig. 3. Let us suppose a population divided into 3 swarms of 3 robots each (Fig. 3a). If swarm 1 and 2 (red and green robots, respectively) cannot improve their objective for SC^{max} iterations they are punished by excluding the worst performing robot of each swarm and adding them to the socially excluded group (Fig. 3b). The socially excluded robots randomly wander in the scenario memorizing their individual best solution and the global best solution of the socially excluded group (Fig. 3c). Swarm 3 improves its solution, since it finds a local optimum, and it is rewarded with the best performing robot in the socially excluded group (Fig. 3d). Finally, the new member of swarm 3 communicates its best individual solution to the other members which is better than their best global solution inducing them to move toward this new solution (Fig. 3e).

Like in *RPSO*, a few parameters also need to be adjusted to run the algorithm efficiently:

- c_1, c_2 and c_3 coefficients for each swarm s;
- Initial, minimum and maximum number of robots in each swarm n_I, n_{min} and n_{max}, respectively;
- Initial, minimum and maximum number of swarms s_I, s_{min} and s_{max}, respectively;
- Stagnancy threshold SC^{max}.

Algorithm I summarizes the *RDPSO* algorithm.

Despite similarities with the *RPSO* algorithm, since this new approach benefit from Darwin's Theory survival of the fittest, it presents a mechanism to escape from local solutions. Furthermore, note that having multiple swarms enables a more distributed approach then in *RPSO* because the network that was previously defined by the whole population of robots is now divided into multiple smaller networks (one for each swarm), thus decreasing the number of nodes (*i.e.*, robots) and the information exchanged between robots of the same network. In other words, robots interaction with other robots through communication is confined to local interactions inside the same group (swarm), thus making *RDPSO* scalable to large populations of robots. Furthermore, since swarms are dynamic (members can be punished or rewarded), there is a larger possibility to escape from local solutions when robots are socially excluded or need to travel from one swarm to another.

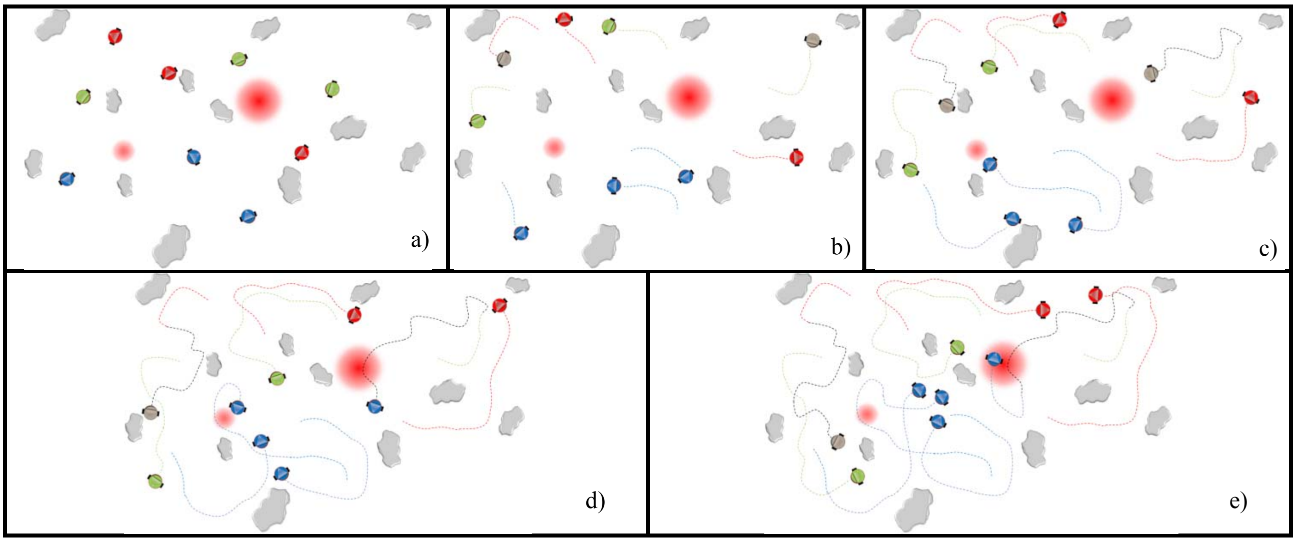

Fig. 3. Sequence of a *MRS* exploration on a scenario with a regular density of obstacles using the *RDPSO* algorithm.

V. Experimental Results

In this section, the effectiveness of using a modified version of *PSO* and *DPSO*, respectively denoted as *RPSO* and *RDPSO*, on a group of agents (*i.e.*, robots) performing distributed unsupervised learning with local and global information is explored. The number of robots is equal to the number of particles in the population, so each robot is represented by a single unique particle. Robots are randomly deployed in the search space. Since both *RPSO* and *RDPSO* are stochastic algorithms, every time they are executed they may lead to different trajectory convergence. Therefore, multiple test groups of 250 trial of 350 iterations each were considered. In the particular case of the *RDPSO*, it is used a minimum, initial and maximum number of 1, 3 and 6 swarms, respectively (represented by different colors in Fig. 3), independently of the population of robots taking into account the algorithm description in section IV where the number of swarms may vary throughout the simulation. In these experiments, the search space is represented by an example of a Gaussian distribution on a function of two variables of the search space, x and y-axis, which represents the position of the robot in meters. The optimum value of this function (-6.54 in the example) is represented in Fig. 5 by a dashed line. Robots will then move in a scenario of size 30 x 30 meters where the z-axis represents the value of the objective function. In this specific case, the objective of the particles is to find the minimum value of the cost. Both algorithms will be evaluated by changing the density of obstacles and the number of robots (*a.k.a.* population) using boxplot charts which is a quick way of examining the final result of each trial graphically. Experiments are then divided into three types: *i*) without obstacles; *ii*) with a regular density of obstacles randomly deployed at each trial; and *iii*) with a high density of obstacles randomly deployed at each trial (*cf.*, Fig. 3). The experimental results obtained without obstacles are used as guidelines for a better understanding of the impact of obstacles in the algorithm's performance since both algorithms performed efficiently without obstacles and obtained the optimal solution. The number of robots will vary from 3 robots to 33 robots with incremental steps of 6 robots, *i.e.*, $N = \{3,9,15,21,27,33\}$ in order to understand the performance of the algorithms related to the population size (Fig. 4). The ends of the blue boxes and the horizontal red line in between correspond to the first and third quartiles and the median values, respectively.

As expected, the rise in the number of obstacles leads to a decrease of performance in both algorithms, for a robot population inferior to 21 robots. It is also clear that the *RPSO* gets stuck in the local optimum (in the neighborhood of 0 and -3), thus increasing the inconsistency of the final result obtained (larger blue boxes and whiskers). This performance gets better as the number of robots rises. It is also verified that for $N \geq 27$ the algorithm tends to stabilize and the impact of the presence of obstacles in the algorithm performance is diminished as robots always arrive at the desired destination. The data distribution, despite the considered trial, turns out to be positively skewed (*i.e.*, the mean is higher than the median). This means that, in this case, as the goal is to minimize the cost function, 50% of the trials are around the desired objective value. Nevertheless, the *RDPSO* shows a better performance when compared with the *RPSO* in the three experimental datasets, being the median (red line) closer to the objective value, regardless of the number of robots.

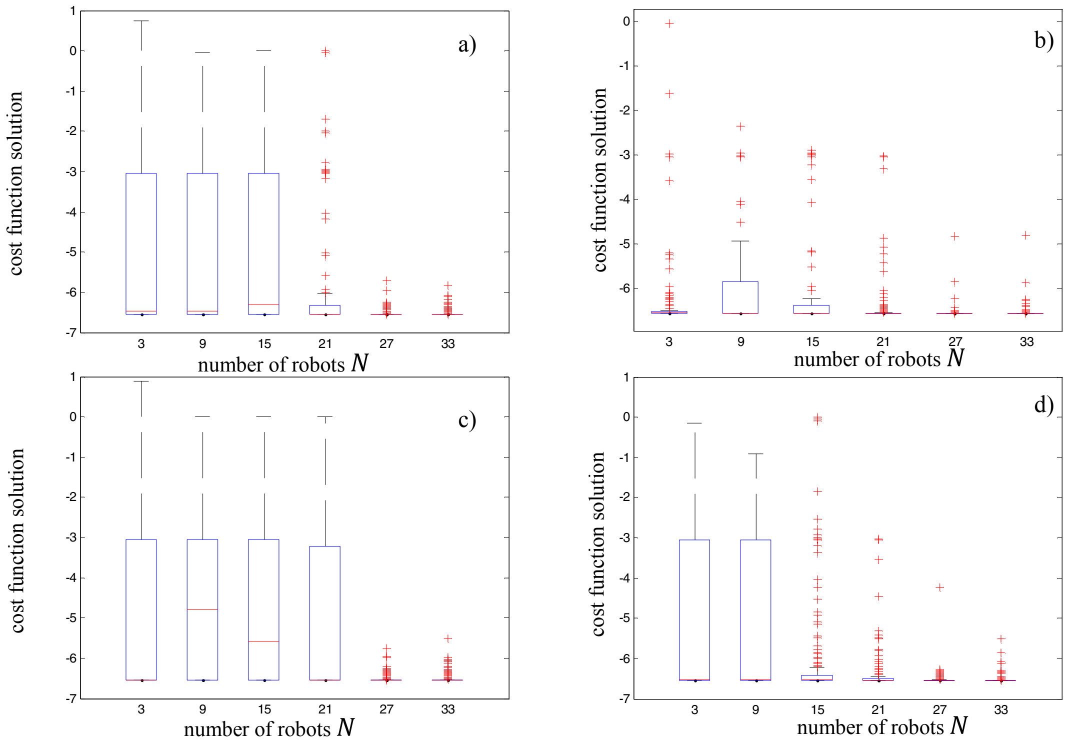

Fig. 4. Performance of the algorithms changing the number of robots N in the population; a) *RPSO* b) *RDPSO* with a regular density of obstacles; c) *RPSO* d) *RDPSO* with a high density of obstacles.

Since these simulation experiments represent a search task, it is necessary to evaluate not only the completeness of the mission but also the speed. Therefore, to further compare both algorithms, the convergence of the *RPSO* and *RDPSO* can be analyzed for the worst case scenario, *i.e.*, for a high density of obstacles. As Fig. 5 shows, the median of the best solution in the 250 simulation was taken as the final output for each value in the set $N = \{3,9,15,21,27,33\}$. Once again, the performance of the *RDPSO* turns out to be better than the performance of the *RPSO*, with a full convergence to the desired objective value at time $t = 50$, regardless of the number of robots considered. This can be explained due to the effect of social exclusion/inclusion described in section IV which main goal is to avoid being stuck in local optima. In the *RPSO* algorithm it is verified that sometimes it gets stuck in local minimum.

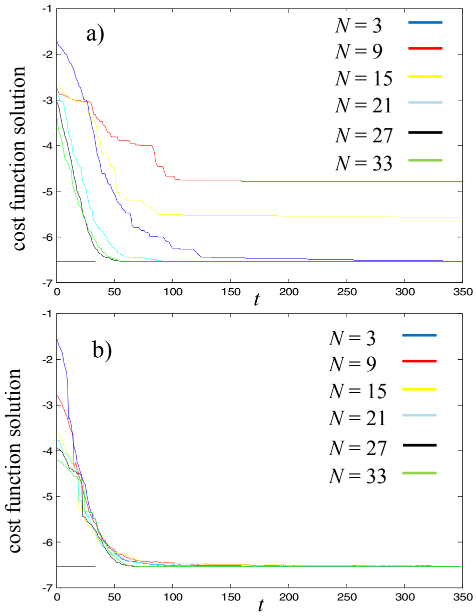

Fig. 5. Convergence of the algorithms changing the number of robots N in the population with a high density of obstacles; a) *RPSO*; b) *RDPSO*.

VI. Conclusion

Modified versions of the Particle Swarm Optimization (*PSO*) and the Darwinian *PSO* (*DPSO*) algorithms based on obstacles avoidance abilities and real-world multi-robot systems (*MRS*) characteristics were developed and respectively named as *RPSO* (Robotic *PSO*) and *RDPSO* (Robotic *DPSO*). The features presented in this document were implemented in a *MatLab* environment and experimental results show how the performance of a *MRS* with a biologically inspired behaviour based on natural selection and social exclusion, as in the *RDPSO*, increases when compared to the *RPSO*. One of the future approaches will be the extension of the *RDPSO* taking into account communication constraints, in a parallel distributed fashion for exploration tasks in *MRS*. Since robots may move to areas of far distance, it is important to have a pervasive networking environment for communications among robots. Furthermore, without a preexistent infrastructure, robots need to be able to act as intermediate nodes in order to relay information from one point to another.

Acknowledgement

This work was supported by a PhD scholarship (SFRH/BD /73382/2010) granted to the first author by the Portuguese Foundation for Science and Technology (FCT) and the Institute of Systems and Robotics (ISR) also under regular funding by FCT.

References

[1] D. Floreano and C. Mattiussi, *Bio-Inspired Artificial Intelligence: Theories, Methods, and Technologies*. Cambridge, MA: MIT Press, 2008.

[2] J. Kennedy and R. Eberhart, "A new optimizer using particle swarm theory," *Proceedings of the IEEE Sixth International Symposium on Micro Machine and Human Science*, pp. 39-43, 1995.

[3] J. Tillett, T. M. Rao, F. Sahin, R. Rao, and S. Brockport, "Darwinian Particle Swarm Optimization," *Proceedings of the 2nd Indian International Conference on Artificial Intelligence*, pp. 1474-1487, 2005.

[4] R. Rocha, J. Dias, and A. Carvalho, "Cooperative Multi-Robot Systems: a study of Visionbased 3-D Mapping using Information Theory," *Robotics and Autonomous Systems*, vol. 53(3–4), pp. 282-311, 2005.

[5] E. Bonabeau, M. Dorigo, and G. Theraulaz, *Swarm Intelligence: From Natural to Artificial Systems*. New York: Oxford University Press, 1999.

[6] H. Q. Min, J. H. Zhu, and X. J. Zheng, "Obstacle Avoidance," in *Proc. Int. Conf. on Machine Learning and Cybernetics*, 2005, pp. 2950-2956.

[7] J. Pugh and A. Martinoli, "Multi-Robot Learning with Particle Swarm Optimization," in *Proceedings of the Fifth International Joint Conference on Autonomous Agents and Multiagent Systems*, 2006.

[8] J. Pugh and A. Martinoli, "Inspiring and Modeling Multi-Robot Search with Particle Swarm Optimization," in *Proceedings of the 2007 IEEE Swarm Intelligence Symposium*, 2007.

[9] H. Çelikkanat and E. Sahin, "Steering self-organized robot flocks through externally guided individuals," *Neural Computing & Applications*, vol. 19, pp. 849-865, 2010.

[10] D. Tardioli, A. R. Mosteo, L. Riazuelo, J. L. Villarroel, and L. Montano, "Enforcing Network Connectivity in Robot Team Missions," *The International Journal of Robotics Research*, 2010.

[11] P. Menezes, "Navegação de Robôs Móveis," MSc Thesis, University of Coimbra, 1999.

[12] K. M. Passino and S. Yurkovich, *Fuzzy Control*. Addison-Wesley, 1998.

[13] C. S. Sahin, et al., "Self-deployment of Mobile Agents in MANETs for Military Applications," *Army Science Conference*, pp. 1-8, 2008.

[14] L. Marques and A. T. d. Almeida, "Finding Odours across Large Search Spaces: A Particle Swarm-Based Approach," in *Proc. 6th Intl. Conf. on Climbing & Walking Robots (CLAWAR)*, Madrid, Spain, 2004.

[15] R. L. Williams and J. Wu, "Dynamic Obstacle Avoidance for an Omnidirectional Mobile Robot," *Journal of Robotics*, no. 901365, p. 14, 2010.

[16] J. Tillett, T. M. Rao, F. Sahin, R. Rao, and S. Brockport, "Darwinian Particle Swarm Optimization," in *Proceedings of the 2nd Indian International Conference on Artificial Intelligence*, 2005, pp. 1474-1487.

[17] T. Burchardt, "Social exclusion: concepts and evidence," in *Breadline Europe: The measurement of poverty*, 2000.

[18] T. Nicholson. (2008) Presentation to Brotherhood of St Laurence's symposium on 'Social Inclusion Down Under'. [Online]. http://www.bsl.org.au/main.asp?PageId=6175